Springboard Capstone Project 2

Foodservice Recommender System

Final Report

Hye Joo Han

1. Project Goal

A food delivery service company (imaginary) asked to analyze ratings for restaurants or food service businesses (e.g., coffee shops and breweries) and build a recommender system. In this project, I wrangle and explore Yelp datasets containing information about foodservice businesses and users and their reviews. Finally, I build a recommender system that predicts star ratings for each user for a given business.

2. Data Wrangling

Collecting data

Yelp datasets were downloaded from https://www.yelp.com/dataset. The datasets include the following 3 files in json format.

1. review.json (review texts, star rating, user_id, business_id, review date and other information for each review)
2. business.json (information about each business in 10 metropolitan areas. Each business information includes location data, number of reviews, average star, attributes, categories and others)
3. user.json (user information including user_id, average stars, first date on Yelp and others)

Cleaning datasets

I cleaned each dataset.

(1) Business dataset

1. I transformed the values of 'categories' column in string format into a list of categories. Among the categories, the top categories related to food service businesses are only 'food' and 'restaurants' categories and others are subcategories.
2. I selected the foodservice businesses using food and restaurant categories. There were originally 188,593 businesses, but the number of businesses was reduced to 72,624.

Missing values

I took care of columns with numeric, string, list and dictionary values separately.

1. Numeric: One missing latitude was filled using its address.
2. String

    a. 3 columns, 'neighborhood', 'address', and 'postal_code' were dropped since they have many missing values and are not likely to be useful in my analysis. Location information is still in latitude and longitude columns.
    b. 3 missing city names were filled using longitude and latitude
3. List: The list columns had no missing value.
4. Dictionary
    a. The column ‘hours’ was removed since it has missing values over 20%.
    b. The ‘attribute’ column with 39 possible business attributes were made into 39 separate columns, but only 4 of them ('BusinessAcceptsCreditCards', 'BusinessParking', 'RestaurantsPriceRange2', 'RestaurantsTakeOut') survived after removing the columns over 20% missing values. Three of the 4 were numerical columns about price range, takeout, and credit card and their missing values were filled with medians. The other column ‘BusinessParking’ had information about parking information in dictionaries with 5 parking types. The column was removed since it is not likely useful in my analysis.

Outliers

All numeric columns except for reivew_count have values in normal range and do not show any suspicious values or outliers. The column 'review_count' has some extremely high values, but it was closely investigated in EDA.

(2) Review dataset

1. All four string columns 'review_id', 'user_id', 'business_id', and 'date' had an actual string value inside "b ", but it was not byte type. (e.g., "b'2011-02-25'" instead of '2011-02-25') Thus, I simply removed "b ". Non numerical columns had no missing values.
2. Numerical columns show some values in a wrong range. The reviews cannot have negative values for 'cool' or 'useful' counts, so I considered -1 for those columns as missing values. There is no zero star option in Yelp ratings, so I considered 0 star as a missing value. There were only 1 row for each missing case, so I simply removed the rows.
3. There were 5,996,996 reviews originally and 4,017,884 reviews were left after cleaning and selecting the reviews for foodservice businesses.

(3) User dataset

1. The numerical columns had no missing values. The distribution of review counts for users is severely right-skewed (some with over 10,000 reviews). I investigated these users in my EDA.
2. All non-numerical columns were string types. There were many empty strings for user names and ‘None’ (string type, not None type) for 'elite' and 'friends' columns. Empty strings are likely to be missing values. However, ‘None' for the 'elite' column is likely to represent users who were never elite members and ‘None’ for the 'friends' column is likely to represent users without friends in Yelp. I did not clean these columns since I was not likely to utilize them.

3. I filtered out the users who have not left reviews for foodservice businesses.

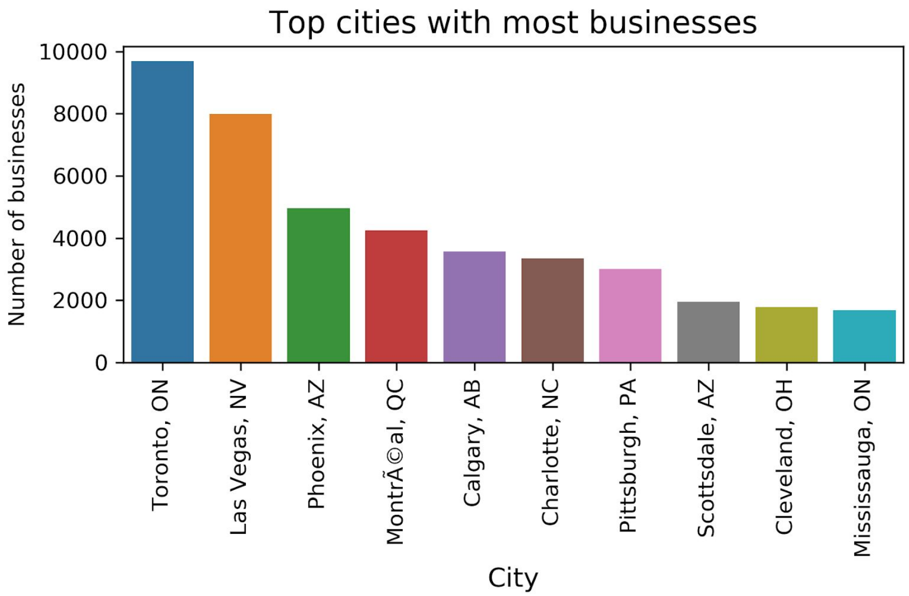
3. Exploratory Data Analysis

Each dataset was explored and the datasets were combined in the end for the recommender system part. The important columns and some relationships between those columns were investigated.

(1) Business dataset

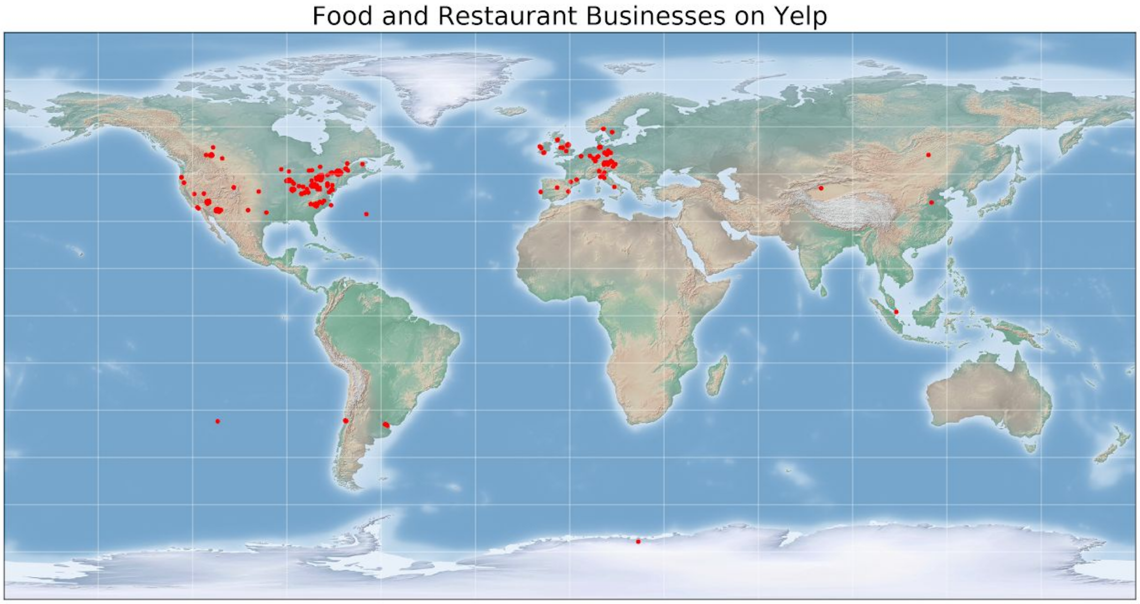
City

Businesses are in 818 different cities, but I found some city names represent the same city and some states have the same name cities. I made a new column with both city and state names to distinguish the same name cities in different states. I cleaned city names only for the two cities Toronto, ON and Las Vegas, NV which have the largest numbers of businesses because I will choose only one metropolitan area eventually for the recommender system. The below figure shows the top ten cities with the most number of businesses.



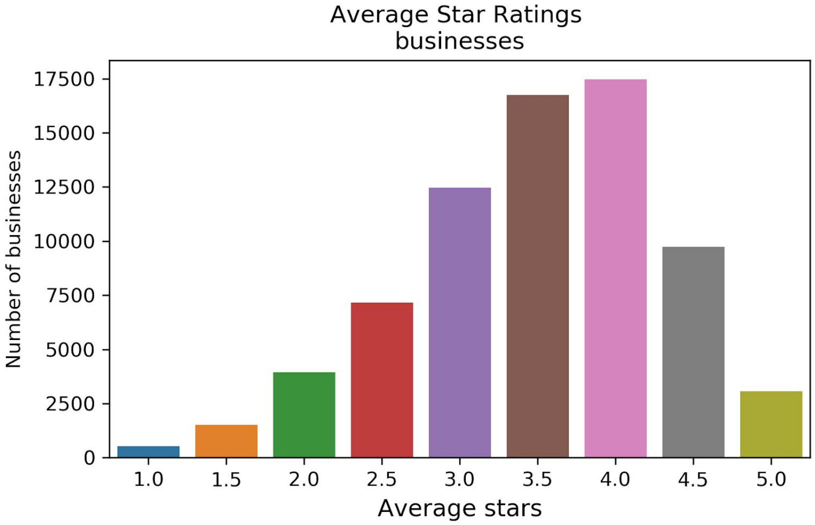
Longitude and Latitude

The world map below showed that most of food and restaurant businesses on this dataset are in North America and Europe and there are a few businesses in South America and Asia. Some dots on the ocean might represent businesses in some islands, but is there even a business in Antarctica? This business was found to be one in Turks and Caicos Islands and I found its swapped latitude and longitude represent the actual location for Turks and Caicos Islands.



Average stars for business

The distribution of average stars is peaked at 4 stars (see below). The number of businesses increases as average stars increase, but drops after star rating 4. This trend might show that businesses with higher stars are more likely to survive and average stars of 4.5 and 5 are difficult for businesses to achieve.



Review count

The distribution of review counts is highly right-skewed. A histogram (not shown here) showed there are 6 businesses with extreme numbers of reviews between 4500 and 8000.

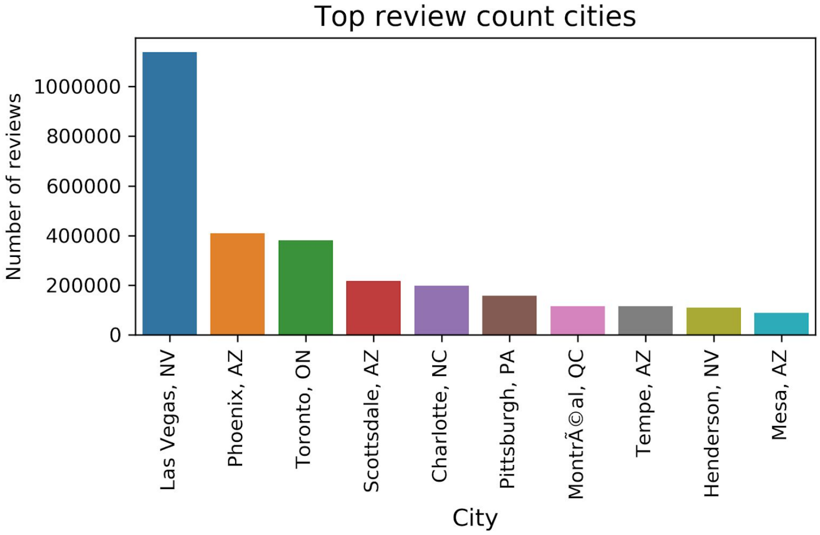
name	restaurant	city_state	stars	PriceRange	review_count
Mon Ami Gabi	1	Las Vegas, NV	4.0	2.0	7968
Bacchanal Buffet	1	Las Vegas, NV	4.0	3.0	7866
Wicked Spoon	1	Las Vegas, NV	3.5	3.0	6446
Gordon Ramsay BurGR	1	Las Vegas, NV	4.0	2.0	5472
Hash House A Go Go	1	Las Vegas, NV	4.0	2.0	5382
Earl of Sandwich	1	Las Vegas, NV	4.5	1.0	4981

All of the 6 businesses with the most number of reviews are found to be restaurants in Las Vegas, NV. Their average stars are all higher than the mean stars (3.49) and their price ranges are higher than the average price range (1.7) except for one.

This result made me wonder three things:

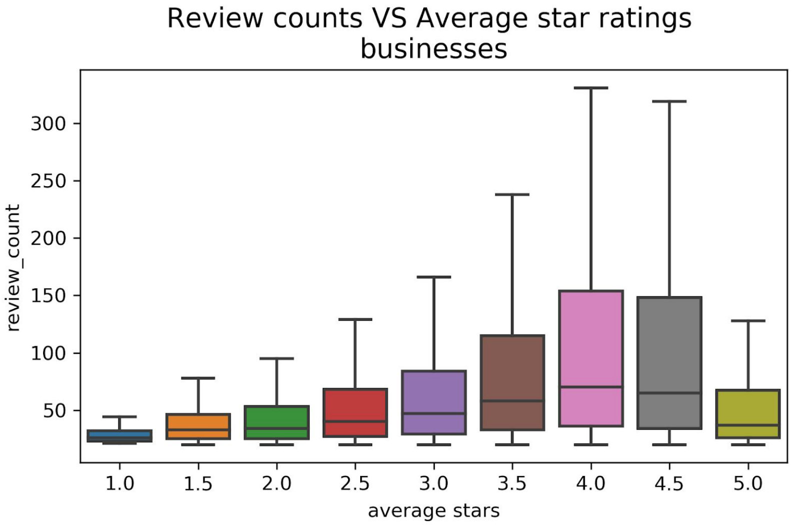
- Which cities have the most reviews
- Whether highly rated businesses (higher stars) tend to get more reviews
- Whether more expensive businesses tend to get more reviews (This question was answered in the price range section)

Which cities have the most reviews?



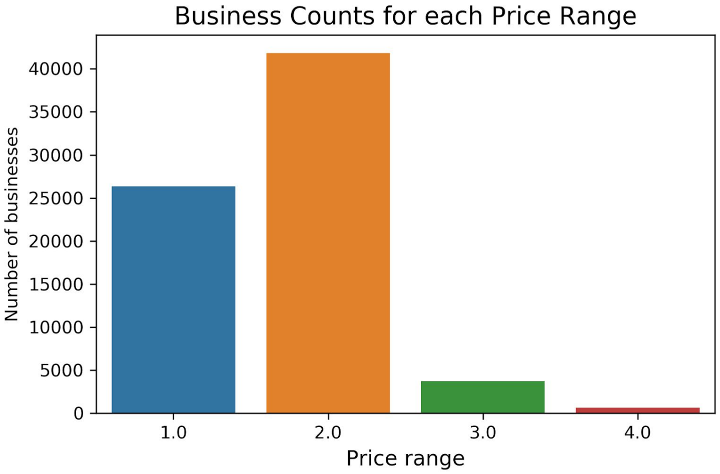
The city with the most number of reviews is Las Vegas, NV. The total review count of Las Vegas (over 1 million) is almost 3 times higher than that of Phoenix, the city with the second most reviews. Note that the 9th city, Henderson is also part of the Las Vegas metropolitan area. Toronto is the third although it has the largest number of foodservice businesses in this dataset.

Do highly rated businesses tend to have more reviews?



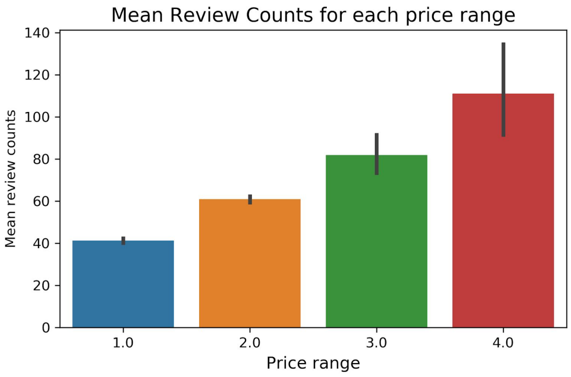
The graph shows that highly rated businesses tend to have more reviews with some exceptions. The review counts decrease after star rating 4 for the average stars 4.5 and 5 (The above boxplots are not showing outliers since they make it hard to see the overall patterns).

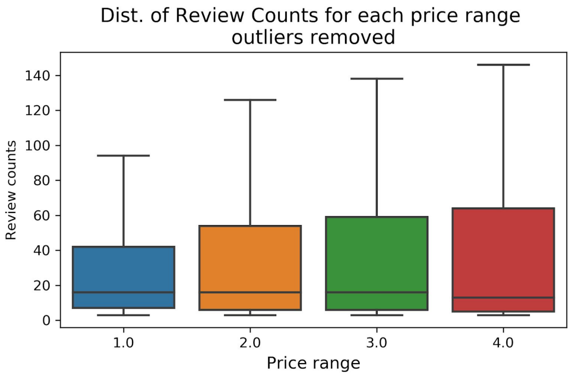
Price range



The most common price range is 2 (same as median) and then 1 and there are much fewer businesses with price range 3 and 4.

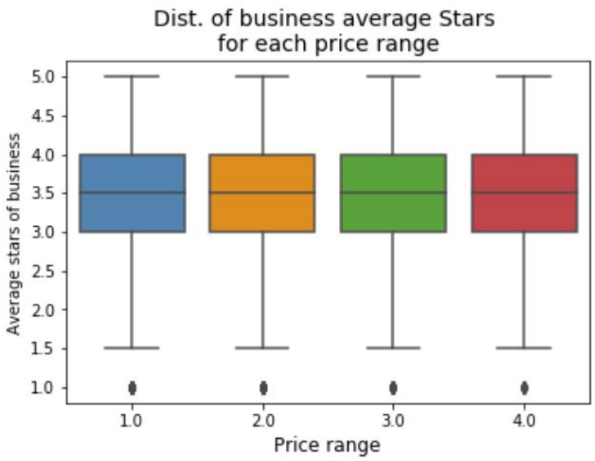
Do more expensive businesses tend to get more reviews?

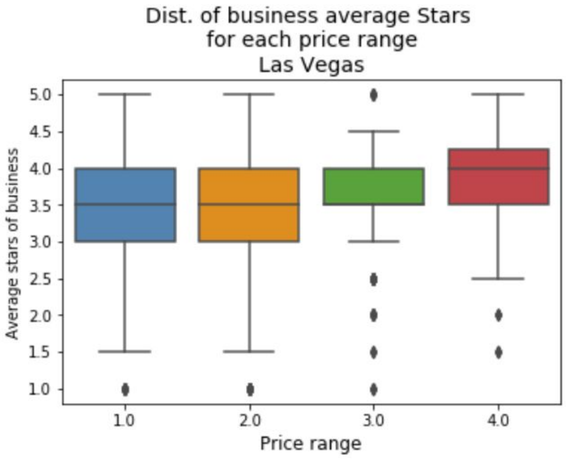




The barplot for mean review counts (left) seems to show people are more likely to leave reviews for more expensive foodservice businesses. However, the boxplot showing the distribution of review counts (right) shows there are not much difference in medians of review counts among the 4 different price ranges (outliers were removed to see the boxes). Moreover, the businesses with the highest price range have the lowest median review counts. Why? This could be because the review counts are highly right-skewed with extremely high review counts as seen in the review count section.

Do more expensive businesses receive higher star ratings?



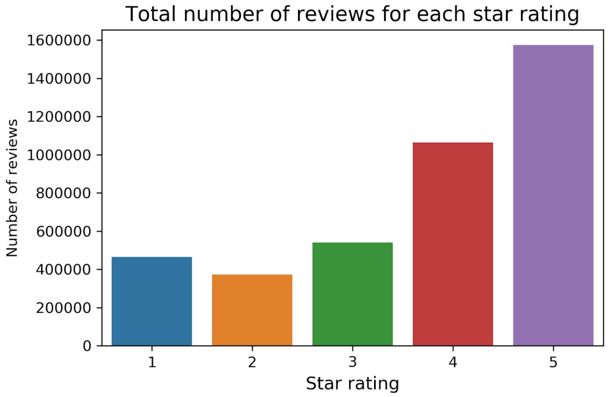


The mean star rating for each price range shows that average stars tend to be slightly higher for more expensive businesses, but the increments are very small ranging from 3.46 to 3.57. The above left figure shows medians and overall distributions of average stars are the same for all price ranges. This might suggest price range is not a good predictor when predicting stars, but there might be some interaction with another feature. For example, the distributions for each price range can vary across different cities (see the right figure). In Las Vegas, the average stars of businesses are distributed differently for different price ranges. Price range 1 and 2 are still similar with median average stars around 3.5, but not price range 3 and 4. For price range 3, average stars tend to be more concentrated between 3.5 and 4 stars. For price range 4, the median star is 4 and the box ranges between 3.5 and 4.2.

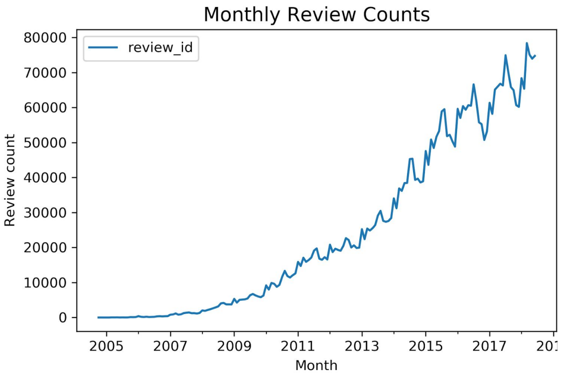
(2) Review dataset

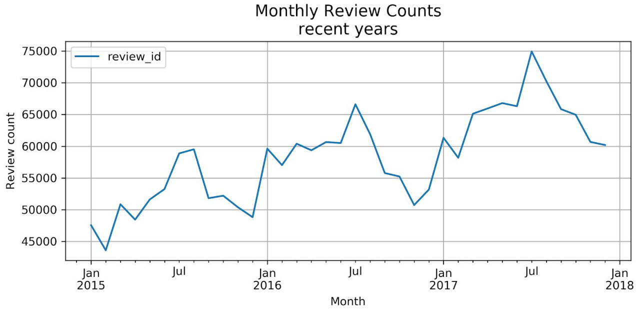
Stars

The below graph shows that reviews with higher ratings are more frequent except for 1 star (1 star is a little more frequent than 2 stars). In the business dataset, we have seen that there are more businesses with higher average stars, but the number of businesses decreases after average star rating 4. If more businesses simply make more reviews for each star rating, the number of reviews as a function star should follow the same pattern. However, 1 star and 5 star do not follow this pattern; 5 stars are more frequent than 4 stars and 1 stars are more frequent than 2 stars. This could mean people tend to leave more reviews than usual when they are highly satisfied (5 star) or very disappointed (1 star).

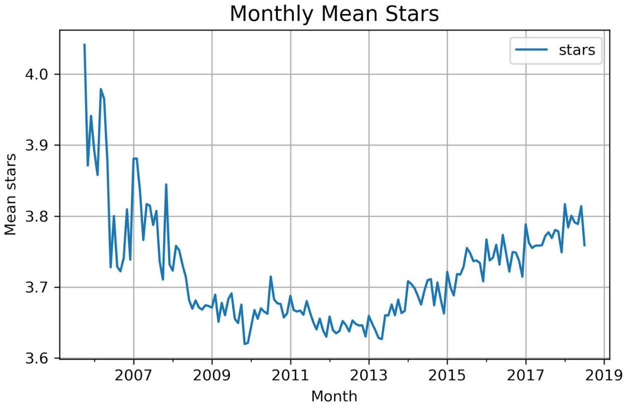


Review date

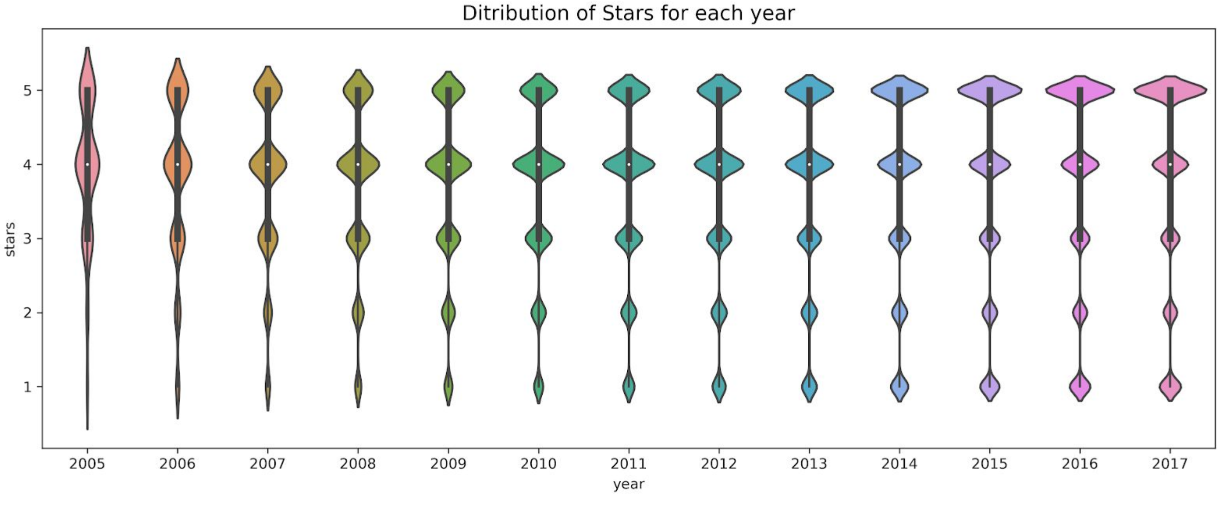




The left graph shows the number of reviews in each month has exponentially increased over time since 2004 and there seems to be seasonal fluctuations. To look into the seasonal pattern more closely, I made another graph (right) using the recent 3 years of reviews. There are indeed seasonal fluctuations. The monthly review count drops during winter months and reaches seasonal peaks around July. People probably eat out (or try new restaurants) more during summer and less during winter.



The above time series graph shows monthly average stars had decreased from around 4 to 3.65 until 2013 and then have increased upto 3.8 till now. This is an interesting and strange pattern, which was more explained in the following graph.



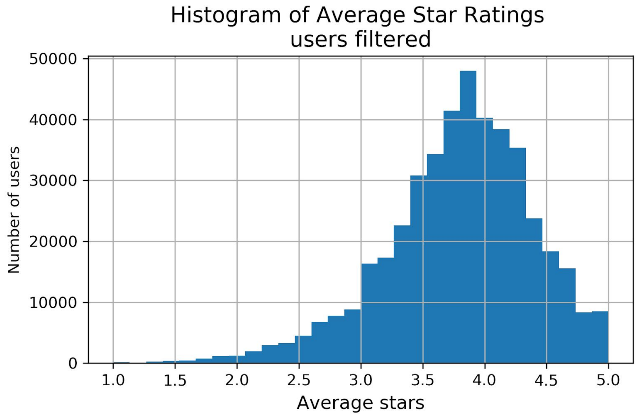
The above yearly violin plots show how distributions of stars changed over time and explain the quadratic shape we saw in the time series graph for monthly mean stars. In the beginning years, low stars are very rare and most stars are 3, 4, or 5. As years go by, 1 or 2 stars also become frequent. This can explain why the average stars were higher in the beginning and decreased over time. Up to 2013, 4 stars are the most frequent star rating, but from 2014 5 stars become the most frequent rating; this can explain the increase of average stars from 2014.

I still do not know why there were fewer 1 and 2 stars in the beginning and more 5's in the recent years. Here are some of my guesses. In the beginning there were only a few businesses and users using Yelp, so users tended not to give harsh ratings. In the recent years, businesses care about their online ratings more and learned to get advices from reviews and receive better stars (e.g., by improving their services or food qualities) .

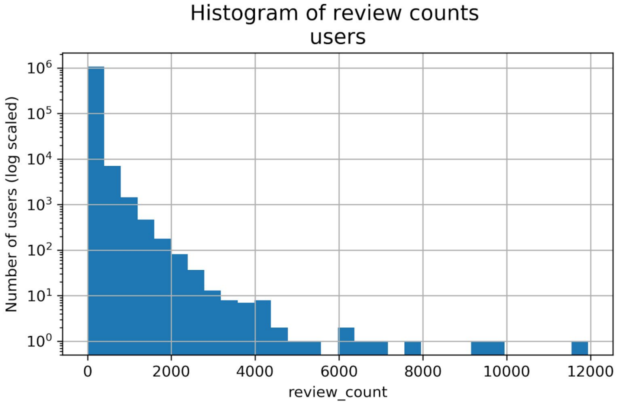
(3) User dataset

Average stars

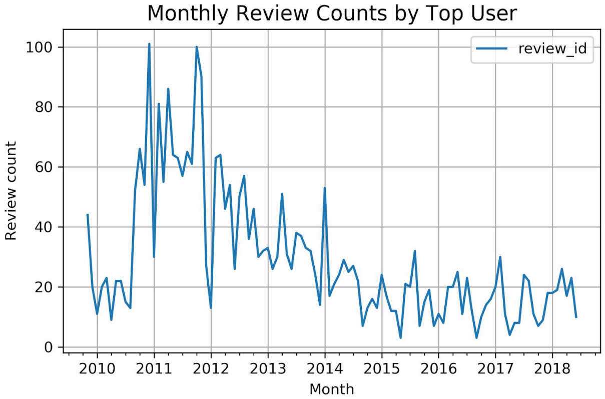
The below graph shows the distribution of user average stars is unimodal with a peak around 3.8 and is left-skewed (I filtered out the users who left less than 10 reviews to remove multiple weird peaks around 1, 2, 2.25, 3, 3.5, 4, 4.25, 4.5, and 5). This is very similar to the distribution of business average stars we saw in the business dataset.



Review count

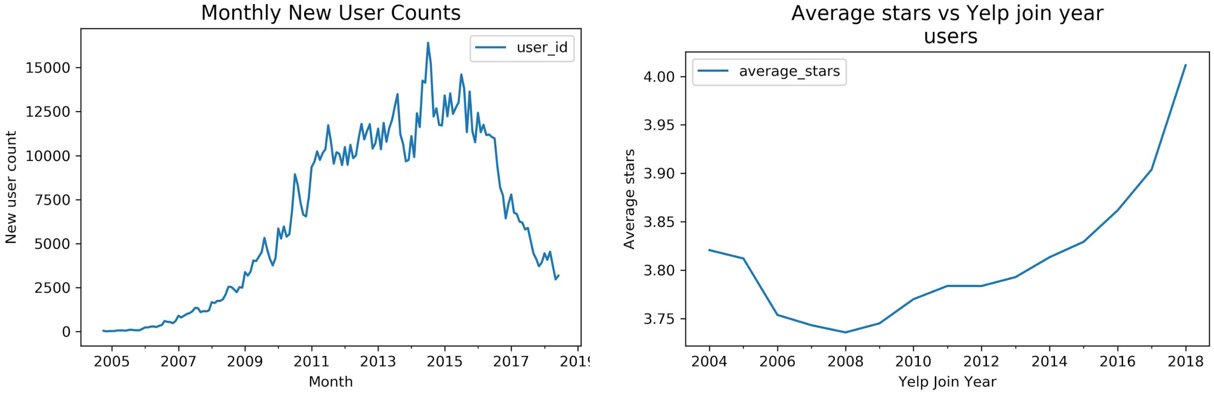


The above log scaled graph shows the distribution of review counts for users is highly right-skewed. There are about 8 people who left even more than 6000 reviews. Who are they? I looked into their user information, but their review counts were not consistent with my review dataset since the originally downloaded datasets are already subsets of all Yelp datasets and I also filtered out non foodservice businesses. Thus, I found the top user from my review dataset instead.



The top user I found has over 3000 reviews for foodservice businesses mostly in cities in Ontario, Canada. The above graph shows the monthly review count of the top user for foodservice businesses. She left around 20 reviews every month and she was even more active between 2010 and 2013. For some months, she left over 100 reviews for foodservice businesses. I am not sure how she can make such a many reviews, but I could not found any other suspicious activities.

Yelping since



Using the Yelp join dates in the ‘yelping_since’ column, I made the monthly new user count graph (left). The graph shows monthly new users increased over time and then started to decrease after the peak around 2014. The dataset does not show who left Yelp, so I was not able to show the number of cumulated users.

The right graph shows the relationship between the Yelp join year and average star ratings of users. The users who joined Yelp earlier tend to have lower average stars ignoring the first couple years. This suggests that Yelp join time (e.g., dates, months, or years since joining) might be a useful user feature when predicting stars.

To use the review dataset as the main dataset, I merged useful columns of business and user datasets into the review dataset.

4. Machine Learning

(1) Preprocessing

Selecting one metropolitan area

To build a recommender system, I selected foodservice businesses only in one metropolitan area for several reasons. First of all, people normally want to have restaurant recommendations in the area they live or plan to visit. Recommending a restaurant in Italy to a person living in Canada without any special reason (e.g. traveling) is not likely to be useful. Moreover, if all cities are used, the matrix by users (rows) and businesses (columns) becomes very sparse and this makes it hard to predict star ratings.

I chose Las Vegas as the target city since it has the most reviews in Yelp. It belongs to the the Las Vegas–Henderson–Paradise metropolitan area in Nevada. Thus, I included all the reviews left for the cities in the metropolitan area. The cities are Henderson, North Las Vegas, Paradise, Las Vegas, and Boulder City. In the EDA part, I already transformed 5 city names all representing Las Vegas into 'Las Vegas'. A city name 'Henderson and Las vegas' actually representing Henderson was also fixed.I further cleaned up names for North Las Vegas, but found no multiple names for Paradise and Boulder City. Selecting the the metropolitan area reviews left 1,280,645 reviews associated with 422,409 users and 9,674 businesses.

Selecting businesses and users with enough reviews

Since it is difficult to predict star ratings for users and businesses with very few reviews, I filtered out users with less than 10 reviews and businesses with less than 20 reviews. The final dataset left has 493,658 reviews associated with 20,340 users and 6,266 businesses. Note that the reviews left could have users with less than 10 reviews and businesses less than 20 reviews because removing users can further reduce the number of reviews for some businesses and vice versa.

Split test and training sets

I set aside 10% of reviews as a test set and used 90% dataset as a training set.

(2) Recommender Systems

I built recommender system models using two types of algorithms, collaborative filtering and content-based filtering. For collaborative filtering algorithms, I used Surprise, a Python package developed for recommender systems. For the content-based filtering algorithms, I built my own models using the basic concept of content-based filtering, but utilized Scikit learn for regressions.

Every model was evaluated using the root mean squared error (RMSE) and RMSEs of all models are reported altogether at the end in order to be compared.

Collaborative Filtering

The collaborative filtering algorithms predict preference or ratings of a user on an item using preference or ratings of other users. These do not utilize metadata of items or users. The followings are some of the algorithms I tried for collaborative filtering. I omitted some algorithms that I did not tune at all due to poor performances or slow speed. I used grid search for hyperparameter tuning (if applicable) and 3 fold cross-validation for every collaborative filtering algorithms.

Normal Predictor

Normal predictor predicts ratings randomly from the normal distribution with mean and standard deviation estimated by the training set. This is a base model to be compared with more complex models.

Baseline

The baseline algorithm predicts ratings using the mean ratings plus user and item biases, which are parameters to be optimized. It also has the regularization term with squares of the biases. By default, the above used Alternating Least Squares (ALS) and I am going to check how Stochastic Gradient Descent (SGD) performs.

SVD

The Singular value decomposition (SVD), a matrix factorization method, is a popular collaborative filtering algorithm done on the user-item rating matrix. The formulas for SVD in the Surprise package are here and also click here for easy explanation.

Co-clustering

This is a collaborative filtering algorithm based on clusters. The users and items are assigned to user clusters and item clusters, respectively, and also to co-clusters which are clustered using both users and items. Then the average ratings of the clusters are used to predict ratings.

Content-based Filtering

The Content-based filtering algorithms use item or user's metadata (e.g., content of films or demographic profile of users) to predict ratings or preferences for recommendations. I built content-based filtering models using the basic concept of content-based filtering . My goal is still predicting star ratings that make the smallest RMSE. To predict stars, I used regression algorithms and the business features as predictor (independent) variables. Since every user has a different taste, model parameters are optimized for each user. I used the same test set I used for collaborative filtering. Cross validation or hyper-parameter tuning for each user was not feasible since many users (almost 3000) have not even 10 business reviews in the training set. I manually tuned hyper-parameters which are the same for all users when necessary. Thus, the test set was used more like a development set in these content-based filtering algorithms and it could have been vulnerable to overfitting.

Data preparation

I used the following business features for the recommender system: city name, latitude, longitude, prince range, review count and categories. The city names and categories had to be further processed since there are 5 different cities (not binary) and a list of categories for each business, respectively; each business could have more than one category due to subcategories. There were originally 439 kinds of categories altogether and I reduced the number to 100 by removing infrequently used categories. Finally, I used one-hot encoding for the city name column and made one column for each category to make binary value columns with 0 or 1.

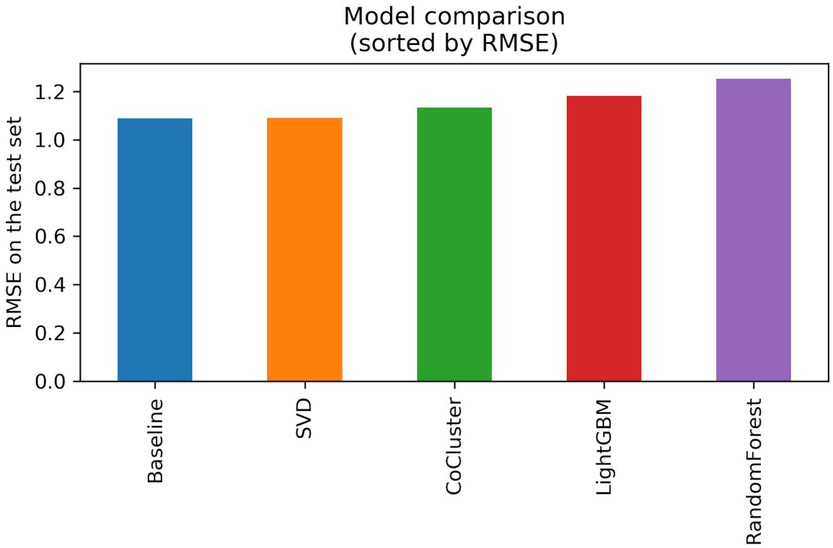
Regression algorithms

Ridge regression
Random Forest
LightGBM

(3) Results

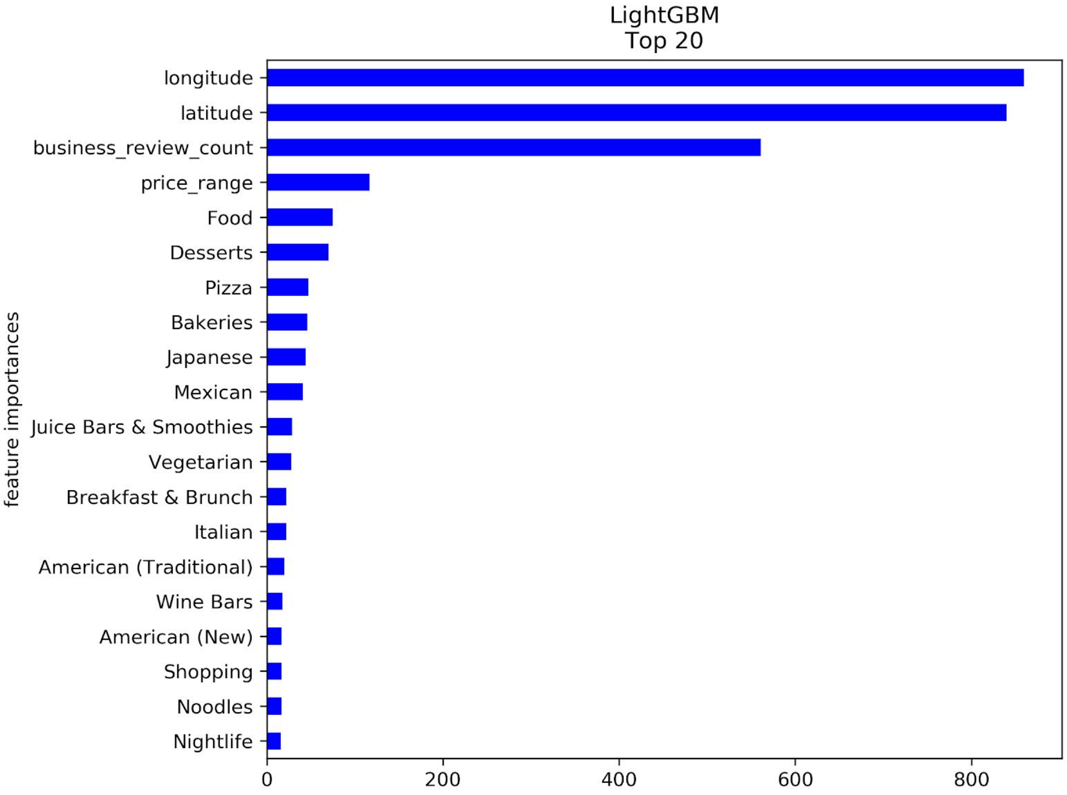
Model comparisons

	RMSE_test	**RMSE_val**
Baseline	1.0891	1.0999
SVD	1.0914	1.1021
CoCluster	1.1332	1.1636
LightGBM	1.1817	1.0950
RandomForest	1.2528	0.4938
Ridge Regression	1.5555	0.7477
Normal Predictor	1.6317	1.6269



Both table and figure are ordered by RMSE on the test set. The top 3 models, Baseline, SVD and Co-clustering, are from collaborative filtering and the 4th and 5th best models, LightGBM and Random Forest, are from content-based filtering.

Feature Importances for the top user



To investigate which features are important, I investigated the feature importances of the best LightGBM model for the user with the most reviews (almost 1400 reviews). LightGBM was chosen since it was the best among the three regression algorithms. The feature importances for the top user showed only 37 features have nonzero importances in the LightGBM model. For this reason, I tried models with smaller numbers of features, but they gave the almost same RMSE and runtime as the original model with all features. It looks like the LGBM model with all features has been doing its best by selecting important features, hence manual feature selection was not necessary. The above figure shows the top 20 important features. The top two features are latitude and longitude. The next two features are the number of reviews and price range.

Future directions

I saw the top user has only RMSE of 0.7629 for the LightGBM model. I would like to further analyze all users' performances and find out which users and businesses made bigger errors.

The content-based filtering models I tried were made only using the basic concept. I would like to try more complex models for content-based filtering and also context-aware collaborative filtering (hybrid of content-based and collaborative filtering).

5. Final Recommendation

The exploratory data analysis found there are much more foodservice businesses with lower price ranges (1 or 2), but businesses with higher price ranges tend to get more reviews on average. Average stars for each business have similar distributions for every price range, but this result does not hold if the same analysis is done for a specific city. For example, Las Vegas showed more expensive food service businesses tend to have higher average stars. Thus, although there are much fewer businesses with high price ranges, they are not less important groups for food delivery services. Moreover, it would be helpful to analyze each city or metropolitan area separately to make a localized marketing strategy.

The number of reviews counted monthly continuously have increased over last 14 years with seasonal fluctuation; people leave more reviews during summer and less during winter. This could suggest that people tend to eat out or try new restaurants more during summer and less during winter. Therefore, summer could be the best season to make profit for food delivery services.

Higher star ratings are more common in reviews overall. In the beginning years, low stars (1 and 2) were very rare, but they became more frequent. Up to 2013, 4 stars were the most frequent star rating, but 5 stars became the most frequent rating since 2014. This could explain the pattern found in monthly average stars, which had decreased until 2013 and then have increased till now. This suggests that one has to apply updated rating standards and consider ratings in recent years rather than just using averages when choosing restaurant partners for delivery services.

The distribution of user average stars is left-skewed and unimodal with a peak around 3.8. Review counts for each user are highly right-skewed and some people left even several thousands of reviews. This suggests that it could be useful to identify elite customers for food delivery services and target those customers with some special marketing strategy.

Las Vegas is the city with the largest number of reviews, so the Las Vegas metropolitan area was chosen to build recommender systems. I tried several different models from either collaborative filtering or content-based filtering algorithms. The top 3 models, Baseline, SVD, and Co-clustering, were all from collaborative filtering algorithms. This suggests that collaborative filtering algorithms would perform better for existing foodservice businesses and users when recommending restaurants that users might like. However, collaborative filtering is not applicable for new businesses or new users (cold start problem) and for such a case, content-based filtering algorithms could come into play. More ideally, context-aware collaborative filtering (hybrid of content-based and collaborative filtering) could be built for better recommendations and new businesses and users.

Python codes with more detailed comments are in the Jupyter notebooks in this link
https://github.com/math470/Springboard_Capstone_Project_2